Adaptive Randomization in Image Steganography Pertaining to Most Significant Nibble

Ravuru Rakesh
SASTRA University, India

Shantan Devathi
SASTRA University, India

Prashanth Sekhar Chandra Sekaran
SASTRA University, India

Siram Sanath Kumar
SASTRA University, India

ABSTRACT
With innovations ruling the world, there has been a phenomenal growth of e-commerce leading to a lot of transactions taking place online in Internet. To safely transfer data, an existing technique called steganography is used. The goal of steganography is to insert a message into a carrier signal so that it can't be detected by unintended recipients. Images, video, and audio being the widespread carrier media steganalysis attempts to discover hidden messages in suspected covers are at the least detectable when they are more randomized. Therefore one of the important considerations in steganography is to have a meliorated randomization technique. To provide better randomization we propose few techniques in image steganography which use adaptive way of randomization by embedding the data in to the bits of "least significant nibble" pertaining to bits in "most significant nibble".

Keywords
Information hiding, Image Steganography, Adaptive randomization

1. INTRODUCTION
World reached the phase where internet becomes the prime mode of communication and subsequently digital crime becomes the major threat to the mankind. Therefore it is essential to have a secured data communication. To gratify this technique called CRYPTOGRAPHY [10-12] evolved which scrambles the data which makes eavesdropper difficult to interpret the data. However the existence of communication is known to the snooper and this may lead to attempts to decrypt, modify or destroy the data. This lead to evaluation of new technique called STEGANOGRAPHY [1-5, 7, 9, 16, 20] which expends data hiding [14, 16, 17].The word steganography was adopted from Greek literature which means "covered writing" [2,3]. The usage of steganographic techniques dates back to prehistoric periods which refer to usage of invisible inks, text on wax-covered tablets, communicating hidden messages in music etc. Little advancements are made in steganography during the period of World War II like Microdot technique [2-4], Communication in Noise of (ISDN-) telephone conversations etc.Recent time advancements in computers, internet, and digital media lead to the wide range development of the steganographic techniques in this domain. It built up as a digital intrigue to hide data in digital media like image files, audio and video etc. The detection of presence of message partially leads to bankruptcy of the system. The strength of steganography can be improved by combining it with cryptography. The two close associates of steganography are watermarking [14, 16] and fingerprinting which are widely used in copyright protection [15]. The main deviation of copyright protection from steganography is the presence of message is often known but robust to remove where as in a steganographic scheme imperceptibility is the crucial factor. A successful eavesdropping attempt to a steganographic scheme is to detect and sometimes retrieve the hidden message where as in copyright protection it is not enough to detect but remove the mark. An efficient steganographic scheme should have high imperceptibility, randomization and capacity.In this paper we propose few novel steganographic schemes in Image steganography [8, 17, 18, 19] which improvise the randomization [20]. The main idea of the schemes stated in this paper is based on the principle of embedding the data in the bits of least significant nibble (LSN) by considering the data bits present in the most significant nibble (MSN). A pictorial representation of MSN and LSN in a byte is shown in Fig 1.

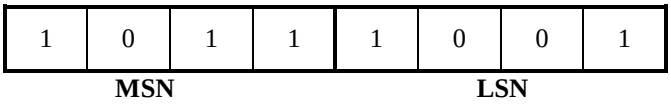

Fig 1. MSN and LSN

2. RELATED WORKS
In the recent past many techniques have been proposed for implementing steganography in images. Security and imperceptibility being the most significant criteria for a successful steganographic scheme, more techniques have been emphasized towards the randomness. The term randomness basically means to shuffle the order of data before embedding in to the cover. The schemes that depends on a key needs to be shared with the recipient, while dealing with the schemes which do not use any key their robustness of the embedding architecture relays on the type of algorithm used. On simple words it refers to the positioning of data in a particular random sequence such as raster [10, 16-18], or any other random scan [13, 21, 22]. The proposed work inherits the randomness through the upper nibble of the respective pixel and scrambles the data accordingly. This actually relays on the basic ideology that the pixel intensities are completely random. So this results in different embedding architectures from image to image and makes the schema more effective by not confining to a single pattern for all the carriers. In extension the present methodology provides techniques towards manipulating the data with reference to the MSN satisfying the important criteria Randomness and security that are most required for any prolific stego method.

3. PROPOSED METHODS

In this section five different randomization methodologies are stated and their algorithms are discussed. Methods 3, 4, 5 have sub methods. Embedding algorithms of the methods always start with reading the image and secret data in to binary and ends with storing the stego image while the extraction algorithms start with reading the stego image in to binary and end with storing the extracted data in required format.

3.1 Method 1 - Ranked nibble embedding

The data in the most significant nibble varies from 0-15. In this method the data is embedded in the LSN's according to the ranked order of MSN's i.e.

a. Data is first embedded in pixels having MSN value 0 and then goes up in series till 15

b. Security could be further enhanced by following random ranking. The randomness could be configured based on a key.

Embedding Algorithm:

1. Get the Image and data in to binary.
2. Get the key and extract the Ranked sequence array.
3. Initialize a variable pointing to the first element of array.
4. Initialize a *loop* and run to traverse entire image.
5. If the value of MSN is equal to variable embed the data
6. End *loop*.
7. Increment the variable pointer to the next value.
8. Go to step 4 if variable is not equal to end value of sequence.
9. Store the image.

Extraction Algorithm:

1. Read the image.
2. Repeat steps 2 and 3 in embedding algorithm.
3. Initialize a *loop* and run to traverse entire image.
4. If the value of MSN is equal to variable extract the data.
5. End *loop*.
6. Increment the variable pointer to the next value.
7. Go to step 3 if variable is not equal to end value of sequence.
8. Store the data in required format.

The Fig 2 illustrates the method of embedding. Each square indicates a pixel and the numbers in the squares indicate data content of MSN in decimal. Black lines indicate the path of embedding and grey lines indicate transition to the next ranked pixel group. In the Fig 2 embedding starts from 0, 1 and 3… (To avoid complexity in figure only first three transitions are marked).

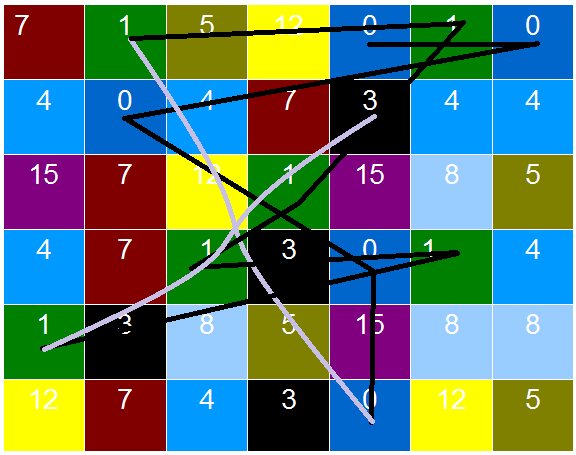

Fig. 2. Ranked nibble embedding

3.2 Method 2 - Embedding in LSN with message XORed MSN

This method uses well known reversible property of XOR to implement keyless random approach. Here each 4 bits of data is XORED with MSN and then substituted in LSN. This scheme provides random adaption of key from the same pixel data that is to be embedded. MSN XORED with LSN retrieves the data and hence no key is to be remembered.

Embedding Algorithm:

1. Initialize a *loop* and run to traverse entire image.
2. Take data bits 4 at a time
3. XOR data bits with MSN and embed in to LSN
4. End *loop*

Extraction Algorithm:

1. Initialize a *loop* and run to traverse entire image.
2. Read the LSN and MSN bits
3. XOR them to get data bits
4. End *loop*
5. Conjoin the data bits and store in required format

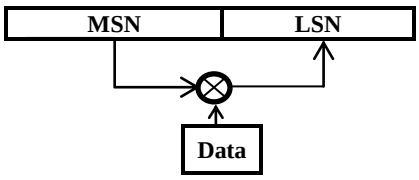

Fig 3. Embedding in LSN

Fig 3 represents embedding algorithm of method 2.

3.3 Method 3 - variable message length embedding

Security can be improved by employing random embedding lengths in each pixel. In this method the number of message bits to be embedded in LSN is varied according to data bits in MSN. The Length of embedding is equal to

1. Number of Ones in MSN or
2. Number of Zeros in MSN or
3. Maximum of number of Ones and Zeros in MSN.

Above three schemes can be used for improvised security of the embedding. Specifically third scheme improves the embedding capacity compared to the above two schemes.

Embedding Algorithm (1, 2):

1. Initialize a *loop* and run to traverse entire image.
2. Count the number of ones/zeros in the MSN
3. Embed the data bits equal to number of ones/zeros in MSN in to LSN
4. End *loop*

Extraction Algorithm (1, 2):

1. Initialize a *loop* and run to traverse entire image.
2. Count the number of ones/zeros in the MSN
3. Extract bits number equal to number of ones/zeros in the MSN
4. End *loop*

Embedding Algorithm (3):

1. Initialize a *loop* and run to traverse entire image.
2. Count the number of ones and zeros and take maximum value
3. Embed the data bits equal to count in to LSN
4. End *loop*

Extraction Algorithm (3):

1. Initialize a *loop* and run to traverse entire image.
2. Count the number of ones and zeros and take maximum value
3. Extract the data bits equal to count in to LSN
4. End *loop*

1	**1**	**0**	**1**	**-**	**E**	**E**	**E**

MSN **LSN**

Fig 4. variable message length embedding

Fig 4 represents embedding algorithm of 1 or 3.'E' represents data is embedded and '-' represents no data is embedded.

3.4 Method 4 - Parity embedding

Parity, a peculiar property of digital data can be used as a key to embed data. In this method the data is embedded in to LSN based on parity of MSN i.e. the number of ones/zeros even or odd. Different sub methods in this are

1. Embed in LSN if number of ones in MSN is even
2. Embed in LSN if number of ones in MSN is odd
3. If number of ones in MSN is even then embed data else if the number of ones is odd embed compliment data

Embedding Algorithm 1:

1. Initialize a *loop* and run to traverse entire image.
2. If number ones in MSN is even embed in LSN else skip
3. End *loop*

Extraction Algorithm 1:

1. Initialize a *loop* and run to traverse entire image.
2. If number ones in MSN is even extract the data from LSN else skip
3. End *loop*

Embedding Algorithm 2:

1. Initialize a *loop* and run to traverse entire image.
2. If number ones in MSN is odd embed in LSN else skip
3. End *loop*

Extraction Algorithm 2:

1. Initialize a *loop* and run to traverse entire image.
2. If number ones in MSN is odd extract the data from LSN else skip
3. End *loop*

Embedding Algorithm 3:

1. Initialize a *loop* and run to traverse entire image.
2. If number ones in MSN is even embed data else if odd embed compliment data in LSN
3. End *loop*

Extraction Algorithm 3:

1. Initialize a *loop* and run to traverse entire image.
2. If number ones in MSN is even extract the data else if odd extract compliment data from LSN
3. End *loop*

3.5 Method 5 - Embed at LSN bits if respective MSN bit is one/zero

To enhance security, variable embedding length property of method 3 can be further extended by appending random positioning of data bits in LSN. This can be achieved by embedding data in the LSN bits

1. If respective MSN bits are one
2. If respective MSN bits are zero

Embedding Algorithm (1, 2):

1. Initialize a *loop* and run to traverse entire image.
2. If MSN bit is one embed data bits at consorting LSN bit positions
3. End *loop*

Embedding Algorithm (1, 2):

1. Initialize a *loop* and run to traverse entire image.
2. If MSN bit is one extract data bits at consorting LSN bit positions
3. End *loop*

1	0	1	1	E	-	E	E

MSN **LSN**

Fig. 5. Parity embedding

Fig 5 represents the embedding algorithm of method 5 (1, 2). 'E' represents data is embedded and '-' represents no data is embedded.

4. STASTICAL ANALYSIS

Data is random in nature. Certain parameters like embedding capacity per pixel can be estimated using the laws of probability. Capacity is the number of bits that can be embedded.Methods 1 and 2 improve the security i.e. randomness and don't affect the capacity and these are similar to a normal LSB substitution method in capacity. But the third method has higher level of secured effectiveness with change in the capacity of data. The fourth & fifth method characterizes similar to the third method. Statistical analysis of embedding capacity is as follows. **B***ij* denotes the embedding capacity per pixel, where *i* specify the number of method and *j* specifies its sub method.

Method 1 & 2

In these two methods full length embedding takes place in LSN and embedding capacity equals to length of nibble i.e.

B*1* = B*2* = 4.

Method 3

For sub methods 1 and 2 the possible number of bits that can be embedded in a pixel is

S = {0, 1, 2, 3, 4}

Probability of each event is 1/5

The average number of bits that can be embedded per pixel is

$$B31 = B32 = \sum_{I=1}^{5} S_I / 5 = 2.$$

For sub method 3, S = {2, 3, 4}. Probability of each event is 1/3

$$B33 = \sum_{I=1}^{3} S_I / 3 = 3.$$

Method 4

For sub methods 1 and 2 the possible number of bits that can be embedded in a pixel is S = {0, 4} and probability of each event is ½. The average number of bits that can be embedded per pixel is

$$B41 = B42 = \sum_{I=1}^{2} S_I / 2 = 2.$$

For sub method 3 full embedding takes place and the average embedding length is

B*43* = 4.

Method 5

The capacity of this method is similar to method 3, sub methods 1 and 2

B*51* = B*52*= 2.

5. SIMULATION AND RESULTS

The simulation for the above discussed methods is implemented in MATLAB for four images 256×256 gray scale TIF images Lena, Baboon, Sagradafamilia, and Eiffel. Peak signal to noise ratio (PSNR) is the parameter used to estimate the quality of stego image i.e. imperceptibility. To calculate PSNR we require a parameter called Mean Square Error (MSE) which is defined as follows

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

m, n represents dimension of image. *I* represents cover image and *K* represents stego image. PSNR is defined as

$$PSNR = 10 \times \log_{10} [\frac{(2^8 - 1)^2}{MSE}]$$

PSNR values are calculated for full length embedding.

Image	**Method1**	**Method2**	**Method3**		**Method4**		**Method5**
			1 / 2	**3**	**1 / 2**	**3**	**1 / 2**
PSNR in dB							
Lena	31.8528	31.8434	37.6262	35.4668	34.8669	31.8552	34.8588
Baboon	31.8447	31.8049	37.5108	35.4372	34.8499	31.8377	34.8595
Sagradafamilia	31.8538	31.8801	37.6469	35.4049	34.8449	31.8391	34.8630
Eiffel	31.8736	31.8613	37.5825	35.5171	34.8670	31.8050	34.8526

Table 1

Fig 6

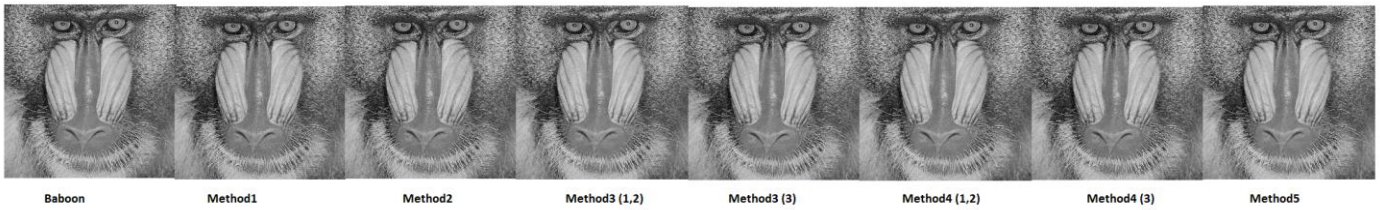

Fig 7

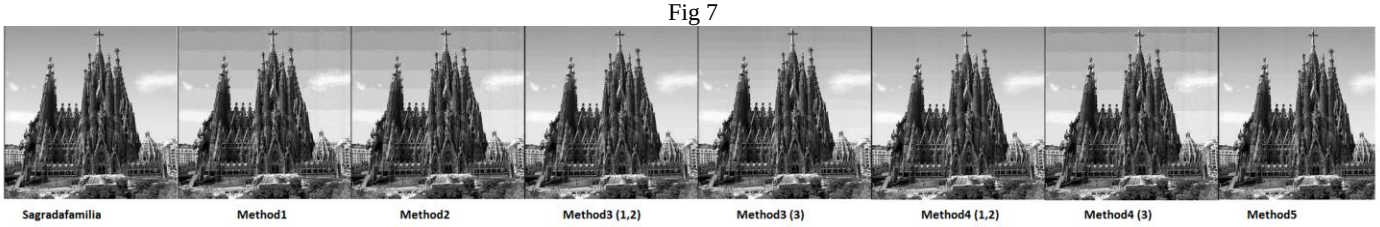

Fig 8

Figures 6, 7, 8 represent cover and stego images oflena.tif, baboon.tif, sagradafamilia.tif.

The peak signal to noise ratio (PSNR) values obtained is well above the threshold of human visual system (30 dB) which implies good imperceptibility. For methods 1 and 2, Table 1 implies the imperceptibility and embedding capacity are same as that of the normal 4-bit LSB substitution method but are more secured than the later one. Methods 3 & 4 (1, 2) compromises between imperceptibility and capacity. Method 3, 4 (3) has the same level of security as that of the previous methods but has increased capacity. Method 5 has a less imperceptibility than the optimal one but has an induced security by adopting bit positioning.

6. CONCLUSION

The main objective of work is to improve the security through adapting therandomization based on a random data i.e. MSN. Few possible methods corresponding to nibble randomization are successfully implemented. These proposed methods have both imperceptibility and induced randomness which gives a tough task to an eavesdropper to break the security walls.

7. ACKNOWLEDGEMENTS

We are thankful to Dr.Juan Antonio Chávez Domínguez, Professor, Sensor Systems Group, UPC, Spain, Balasurendrakumar, Manikanta Chaitanya,students of ECE,AmirtharajanRengarajan, Assistant professor and John BoscoBalaguru, Associate dean-Research ofSASTRA University for their support.

8. REFERENCES

[1] Moerland, T., “Steganography and Steganalysis”, Leiden Institute of Advanced Computing Science,

[2] Silman, J., “Steganography and Steganalysis: An Overview”, SANS Institute, 2001

[3] Jamil, T., “Steganography: The art of hiding information is plain sight”, IEEE Potentials, 18:01, 1999

[4] Wang, H & Wang, S, “Cyber warfare: Steganography vs. Steganalysis”, Communications of the ACM, 47:10, October 2004

[5] Anderson, R.J. &Petitcolas, F.A.P., “On the limits of steganography”, IEEE Journal of selected Areas in Communications, May 1998

[6] Marvel, L.M., Boncelet Jr., C.G. &Retter, C., “Spread Spectrum Steganography”, IEEE Transactions on image processing, 8:08, 1999

[7] Johnson, N.F. &Jajodia, S., “Exploring Steganography: Seeing the Unseen”, Computer Journal, February 1998

[8] C.C. Thien, J.C. Lin, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, Pattern Recognition 36 (11) (2003) 2875–2881.

[9] Krenn, R., “Steganography and Steganalysis”,

[10] BruiceSchneier, Applied Cryptography Protocols, Algorithm and Source Code in C. Second edition. Wiley India edition 2007

[11] W. Diffie and M. E. Hellman, ―Exhaustive Cryptanalysis of the NBS Data Encryption Standard,‖ IEEE Computer, Vol.10, 1977, pp. 74-84.

[12] R. L. Rivest, A. Shamir, and L. Adleman, ―A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, vol. 21, no. 2, (1978) 120–126.

[13] Westfeld Space filling curves in steganalysis in E.J Delp III & P.W. Wong (Eds), Security, Steganography and watermarking of multimedia contents VII SPIE 5681, (2005) 28-37.

[14] F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding—a survey, Proc. IEEE 87 (7) (1999) 1062–1078.

[15] S. Katzenbeisser, F.A.P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Norwood, MA, 2000.

[16] W. Bender, D. Gruhl, N. Morimoto, A. Lu, Techniques for data hiding, IBM Syst. J. 35 (3&4) (1996) 313–336.

[17] Yuan-Hui Yu , Chin-Chen Chang, Iuon-Chang Lin, A new steganographic method for color and grayscale image hiding, Computer Vision and Image Understanding 107 (2007) 183–194

[18] C.K. Chan, L.M. Chen, Hiding data in images by simple LSB substitution, Pattern Recognition 37 (3) (2004) 469–474.

[19] R.Z. Wang, C.F. Lin, J.C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recognition 34 (3) (2000) 671–683.

[20] Adnan Gutub, "Pixel Indicator Technique for RGB Image Steganography", Journal of Emerging Technologies in Web Intelligence (JETWI) (2010)2(1) 56-64

[21] Hans Sagan, Space-Filling Curves, Springer-Verlag, New York, (1994). ISBN: 0-387-94265-3

[22] Provos, N., Honeyman, P, Hide and seek: An introduction to steganography, IEEE Security & Privacy Magazine 1 (2003) 32-44.